UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISION

ORACLE AMERICA, INC. Plaintiff, v. GOOGLE, INC. Defendant.	Case No. 3:10-cv-03561-WHA

REPLY EXPERT REPORT OF ANDREW HALL

TABLE OF CONTENTS

I. INTRODUCTION

A. Case Background

1. I have been asked by Google to review and respond to the Rebuttal Expert Report of Gwyn Firth Murray ("Murray Report") and Section VIII of the Expert Report of Chris F. Kemerer, Ph.D., Regarding Fair Use and Rebuttal to Google's Opening Expert Reports dated February ("Kemerer Rebuttal Report"), both dated February 8, 2016.

2. My qualifications and my compensation are set forth in paragraphs 1, 5-12 and Appendix A of my Opening Report, which I incorporate herein by reference.

3. At this time, I have not created any demonstrative exhibits to be used as a summary of, or as support for, my opinions, setting aside the graphics in this report. I reserve the right to create any additional summaries, tutorials, demonstrations, charts, drawings, tables, and/or animations that may be appropriate to supplement and demonstrate my opinions if I am asked to testify at trial.

4. I understand that discovery is ongoing in this case. I therefore reserve the right to supplement my opinions after I have had the opportunity to review deposition testimony or in light of additional documents that may be brought to my attention. If Oracle or its experts change their opinions (either explicitly or implicitly) in such a manner as to affect my conclusions, I may supplement my opinions.

B. Materials Considered

5. In forming my opinions and preparing this reply report, I have considered the materials cited in both the Opening Expert Report of Andrew Hall (Jan. 8, 2016) ("Opening Report") and Rebuttal Expert Report of Andrew Hall (Feb. 8, 2016) ("Rebuttal Report")

(including those materials listed in Appendices B and D, respectively), the materials cited and listed in this Reply Report, as well as the documents listed in Appendix E.[1]

6. I also have specialized knowledge of the matters set forth herein, and if called as a witness I could and would testify competently thereto.

7. My research and analysis of the materials, documents, allegations, and other facts in this case are ongoing, and if additional information becomes available through discovery and depositions in this action, I reserve the right to provide additional opinions.

II. SUMMARY OF OPINIONS

8. Ms. Murray concedes that OEM distributions of the OpenJDK-based Java API packages[2] fall within the Classpath Exception of the GPL-2.0-CE license and, as a result, only OEMs ***modifying*** the OpenJDK-based Java API packages will be subject to the GPL-2.0-CE license's copyleft requirements with respect to those modifications.

9. Furthermore, based on my review of the Murray Report, I understand that Ms. Murray does not: (a) opine that Google's inclusion of the OpenJDK-based Java API packages in the Android stack ***requires*** any software ***other than*** the OpenJDK-based Java API packages to be published under the GPL-2.0-CE license; (b) identify any specific source code combinations that ***would*** require software ***other than*** the OpenJDK-based Java API packages (and modifications thereof) to be published under the GPL-2.0-CE license; or (c) opine that Google's distribution of the OpenJDK-based Java API packages violates the terms of the GPL-2.0-CE license. To the contrary, Ms. Murray concedes she "cannot state unequivocally that

[1] As in my Rebuttal Report, I am starting the lettering of appendices to this reply report at E in order to avoid having more than one Hall Appendix with the same letter designation.

[2] I used this terminology in my Opening Report to describe the 37 Java API packages that Google is now using from the OpenJDK project in place of the previously used same 37 Java API packages from the Apache Harmony project (defined as the "accused Harmony-based Java API packages" in my Opening Report at ¶ 106).

Google's anticipated inclusion of OpenJDK-based code into its core libraries would or would not conform to the specific terms of the GPLv2-CE."[3]

10. Ms. Murray opines that Google's distribution of the OpenJDK-based Java API packages in source code form "could create a significant risk that additional source code, not currently covered by the GPLv2-CE, would have to be distributed under that license."[4] However, in my opinion, Ms. Murray misinterprets the requirements of the GPL-2.0 and GPL-2.0-CE licenses in concluding that Google's distribution "***could*** create a significant risk" because the copyleft requirements of the GPL-2.0-CE license do not differ depending on the format in which the software is distributed (e.g., whether the software is distributed as source code or as an executable (aka, "binary")) and are limited to ***modifications*** to the GPL-2.0-CE software, ***regardless*** of whether the Classpath Exception applies to source code distributions of GPL-2.0-CE software. Indeed, Oracle's own distribution of independent and separate but aggregated source code files under the GPL-2.0-CE, GPL-2.0, and Apache 2.0 licenses indicates that Oracle itself does not apply Ms. Murray's overly broad interpretation of the GPL-2.0-CE license's copyleft requirements.

11. Ms. Murray alleges that widespread confusion and uncertainty exist regarding the copyleft requirements of the GPL-2.0 and GPL-2.0-CE licenses, but her cited references do not support that view. Instead, the references describe companies making informed decisions based on the known and knowable requirements imposed by the GPL-2.0 and GPL-2.0-CE licenses.

12. Ms. Murray conflates the GPL and LGPL licenses with the GPL-2.0-CE license, relying on statements made regarding requirements unique to the GPL and LGPL licenses to draw conclusions regarding the desirability of the GPL-2.0-CE license. Ms. Murray also

[3] Murray Report ¶ 142.
[4] Murray Report ¶ 21.

confuses Google's selection of the accused Harmony-based Java API packages to be an indication that Google found all other solutions unacceptable. Relying upon on this logical fallacy, Ms. Murray expresses doubts regarding whether Google would have been willing to include software licensed under the GPL-2.0-CE license without even acknowledging or addressing that Google ***already*** includes—and has ***always*** included—software licensed under the more restrictive GPL and LGPL licenses as part of its Android distribution.

13. For the reasons explained above, I disagree with Ms. Murray's conclusions regarding the potential scope of the GPL-2.0-CE license's copyleft requirements and that Google and/or its partners would not have accepted licensed alternatives to the 37 Java API packages in the 2007-2010 timeframe. To the contrary, it is my opinion that Google and its partners would not have had significant or lasting concerns in 2007 (or later years) if the 37 Java API packages in Android had been licensed under the GPL-2.0-CE license.

14. I also disagree with Dr. Kemerer's opinion that "Google did not use and would not have used the OpenJDK code in approximately 2007, given the risks of the GPLv2 with Classpath Exception license" and that "Google would not have used that code due to the significant business and legal risk."[5] As explained in my Opening Report, "Google could have licensed and modified the 37 API packages from OpenJDK (as opposed to Apache Harmony) when OpenJDK was first released under the GPL-2.0-CE license as of 2007."[6] Furthermore, as explained in my Rebuttal Report, which focused on opinions very similar to those Dr. Kemerer repeats in his Rebuttal Report, "had Google chosen to adopt GPL-2.0-CE licensed OpenJDK

[5] Kemerer Rebuttal Report ¶ 230.
[6] Hall Opening Report ¶ 27.

class libraries for use in Android, the adoption of Android by OEMs would not have been impeded."[7]

III. DETAILED OPINIONS IN REBUTTAL TO MS. MURRAY'S REPORT

A. Ms. Murray concedes that proprietary OEM software distributed with the OpenJDK-based Java API packages will fall within the Classpath Exception.

15. Ms. Murray concedes that (a) the OpenJDK-based Java API packages included in OEM distributions of Android will fall within the Classpath Exception of the GPL-2.0-CE license and (b) that OEMs will be subject to the copyleft requirements of the GPL-2.0-CE license ***only if*** such OEMs ***modify*** the OpenJDK-based Java API packages:

> It is my opinion that ***OEMs are permitted under GPLv2-CE*** to use the code they receive from Android—even if under GPLv2-CE—to form an executable by linking the GPLv2-CE code to their own independent modules, create a resulting executable, and then distribute that executable under the terms of their choice. However ***if OEMs modify any elements that are subject to the GPLv2-CE license***, they risk being required to distribute those modifications (and the resulting whole work) under the terms of the GPLv2-CE in order to comply with the license.[8]

16. Ms. Murray's interpretation supports my opinion that the inclusion of the GPL-2.0-CE-licensed libraries "would impact only those OEMs that both (a) desire to modify the OpenJDK class libraries and (b) object to publishing those modifications in source code form under the GPL-2.0-CE license."[9] As detailed in my Opening and Rebuttal Reports and explained by Anwar Ghuloum, the Google Director of Engineering responsible for relevant portions of the Android platform, OEMs add components to other portions of the Android software stack but are generally unlikely to modify the OpenJDK-based API packages.[10]

---

[7] Hall Rebuttal Report ¶ 9.
[8] Murray Report ¶ 174 (***emphasis*** added).
[9] Hall Rebuttal Report ¶ 24.
[10] Ghuloum Depo. at 36:11-37:10; 37:12-19. I have reviewed paragraphs 292 through 303 of Dr. Schmidt's rebuttal report, in which he explains how he concluded, based on an analysis of decompiled code, that HTC allegedly modified 24 of the source code files for the 37 API packages, and that LG allegedly modified 60 of those source code files. Notably, Dr. Schmidt offers no opinion that any of these changes are material, haven't already been

Accordingly, it is my opinion that OEMs likely either (a) will not modify the OpenJDK-based Java API packages included in the Android stack or (b) will not be opposed to publishing those modifications in source code form under the GPL-2.0-CE license.

B. Ms. Murray does not opine that Google's distribution of OpenJDK-based Java API packages as part of Android violates the terms of the GPL-2.0-CE license or taints other independent and separate but aggregated code.

17. Notably, Ms. Murray does not: (a) opine that Google's inclusion of the OpenJDK-based Java API packages in the Android stack ***requires*** any software ***other than*** the OpenJDK-based Java API packages to be published under the GPL-2.0-CE license; (b) identify any specific source code combinations that could require software ***other than*** the OpenJDK-based Java API packages to be published under the GPL-2.0-CE license; or (c) opine that Google's distribution of the OpenJDK-based Java API packages violates the terms of the GPL-2.0-CE license. To the contrary, Ms. Murray concedes she "cannot state unequivocally that Google's anticipated inclusion of OpenJDK-based code into its core libraries would or would not conform to the specific terms of the GPLv2-CE."[11]

C. Ms. Murray misapplies her technical understanding based on Dr. Schmidt's Rebuttal Report.

18. In paragraphs 169-176 of her report, Ms. Murray cites the February 8, 2016 Expert Report of Professor Doug Schmidt (the "Schmidt Rebuttal Report") as the basis of her understanding of numerous technical aspects of Google's distribution of Android source code twelve times. In each case, Ms. Murray fails to identify which portions of the 150-page

---

contributed back to the Android code base, or that they would allow OEMs to differentiate their products. In many instances, the alleged changes identified by Dr. Schmidt are trivial—e.g. removing code and then adding it back with only minor changes. *See* Schmidt Rebuttal Report, Table 16 ("Removes a block of implementing code and re-adds it with slightly different argument names and values").

[11] Murray Report ¶ 142.

Schmidt Rebuttal Report she found relevant to her technical understanding (such as pages, paragraphs, or sections).[12]

19. For example, Ms. Murray cites to the Schmidt Rebuttal Report in paragraph 175 where she explains: "I understand from Dr. Schmidt's report that an analysis of OEM Android devices indicates that OEMs have modified the 37 Java API packages that Google now has made available under GPLv2-CE." However, Ms. Murray appears to misunderstand Dr. Schmidt's explanation for why his referred-to evaluation of Android did not include versions of Android incorporating the OpenJDK-based Java API packages:

> I am not aware of any OEMs who are currently distributing software based on the source code in the AOSP Master Branch (which includes copied source code of the 37 Java API packages from OpenJDK 8). Nonetheless, I analyzed the modifications OEMs make to prior versions of AOSP code based on the code I found on OEM devices.[13]

As noted above, Ms. Murray concedes that an OEM distributing the OpenJDK-based API packages will be subject to the copyleft requirements of the GPL-2.0-CE license only if that OEM modifies the OpenJDK-based Java API packages. As such, Ms. Murray should have recognized that evidence regarding alleged modifications regarding one code base subject to an Apache-2.0 license would not necessarily be representative of modifications that OEMs might want to make to a different code based under a GPL-2.0-CE license. Moreover, I note that Dr. Schmidt offers no opinion that any of the changes allegedly made by OEMs to the Java API packages are material, haven't already been contributed back to the Android code base, or are likely to allow those OEMs to differentiate their products from those of other OEMs.

20. I note separately that Ms. Murray erroneously criticizes my statement that "The GPL-2.0-CE license is distinguishable from many other weak-copyleft licenses in that the

[12] Murray Report ¶ 169-177 (nn. 126-37).
[13] Schmidt Rebuttal Report ¶ 21.

GPL-2.0-CE license exempts those qualifying for the Classpath Exception from any compliance obligations related to the GPL-2.0-CE-licensed software."[14] In context, my point was and remains clear: the Classpath Exception exempts ***qualifying code*** that simply ***links to*** code under the GPL-2.0-CE license. Indeed, that is precisely what I explained:

> In fact, the only obligation placed on distributors ***qualifying for the Classpath Exception*** is compliance with the requirements imposed by whichever other license(s) may be covering the ***linking*** GPL-incompatible software. By contrast, many other weak-copyleft licenses (such as the LGPL, MPL, and CDDL) obligate software distributors to offer or provide recipients with the source code corresponding to the distributed weak-copyleft packages under the same license. Weak-copyleft licenses often obligate distributors to provide recipients with additional notices such as warranty disclaimers, a copy of the applicable weak-copyleft license, or other licensing information. By releasing ***qualifying software distributors*** from any obligation with respect to the GPL-2.0-CE-licensed software, the GPL-2.0-CE license places fewer obligations on qualifying distributors than many permissive open-source licenses.[15]

D. The requirements of the GPL-2.0-CE license are the same regardless of whether the software is distributed in source code or executable form.

21. As detailed above, Ms. Murray concedes that the distribution of the OpenJDK-based Java API packages linked with Android in executable form qualifies for the Classpath Exception of the GPL-2.0-CE license and that, as a result, the copyleft requirements of the GPL-2.0-CE license extend only to modifications to the OpenJDK-based Java API packages by OEMs.[16] However, Ms. Murray misunderstands the GPL-2.0-CE license's copyleft requirements for source code distributions, opining that distributions of the same combination of software in source code form "could create a significant risk that additional source code, not currently covered by the GPLv2-CE, would have to be distributed under that license."[17]

---

[14] Murray Report ¶¶ 160, 162 (quoting Hall Opening Report ¶ 61).
[15] Hall Opening Report ¶ 61.
[16] Murray Report ¶ 174.
[17] Murray Report ¶ 21 (emphasis added); *see also* ¶ 142 ("based on my analysis of the GPLv2-CE, and my understanding of how Google has implemented the 37 class libraries at issue to date, I believe that Google ***could significantly risk exposing*** other portions of the Android stack to the copyleft provisions of the GPLv2 by using the OpenJDK code in its core libraries."); ¶ 164 ("[I]t is my opinion that Google's incorporation of OpenJDK-based

22. Ms. Murray bases her equivocal conclusion on her foundational opinions that (a) the Classpath Exception does not extend to source code distributions of GPL-2.0-CE software and (b) the copyleft requirements of the GPL-2.0 license extend to the "whole work." In my opinion, Ms. Murray misinterprets or misunderstands the requirements of the GPL-2.0 and GPL-2.0-CE licenses when she asserts that they ***could*** be different for source code and executable distributions of the same OpenJDK-based Java API packages.

23. As detailed in my Opening Report and Rebuttal Report and summarized in the remainder of this section, in my opinion: (a) the copyleft requirements of the GPL-2.0 license extend to distributed ***derivative works*** of GPL-2.0-licensed software; (b) the copyleft requirements of the GPL-2.0-CE license do not differ in scope based on whether the licensed software is distributed as source code or as executable files; and (c) the copyleft requirements on Google's distribution of the OpenJDK-based source code files is limited to ***modifications*** to the GPL-2.0-CE-licensed software, regardless of whether the Classpath Exception applies to that distribution.

1. The GPL-2.0 license limits its copyleft requirements to "derivative works" of the GPL-2.0 software.

24. In paragraphs 158, 168, and 172 of her report, among others, Ms. Murray interprets the strong-copyleft GPL-2.0 license to extend its copyleft requirements to the "whole work" that includes GPL-2.0-licensed source code.[18] As explained in my Opening Report and Rebuttal Report and summarized in the remainder of this subsection, the GPL-2.0 license clearly limits its copyleft requirements to distributed ***derivative works*** of the licensed software.

---

code in the master branch of Android ***poses a significant risk*** that additional elements of the Android stack (including modifications made by OEMs) would be subject to the requirements of GPLv2-CE."); ¶ 167 ("It is my opinion that ***there is a significant risk*** that GPLv2-CE requires source code of at least some elements of the Android stack … to be distributed under the GPLv2-CE license … .").

[18] Murray Report ¶ 168 ("Mr. Hall does not address the additional requirements of Section 2—that any source code distributions of ***any work containing*** the Program ***must be licensed as a whole*** under the GPLv2.")

25. A derivative work is a work ***based on*** or ***derived from*** one or more already existing works.[19] The GPL-2.0 license distinguishes between the "Program" licensed under the GPL-2.0 license and a "'work ***based on*** the Program' [which] means either the Program or ***any derivative work under copyright law***[.]"[20] Section 2 of the GPL-2.0 license, which Ms. Murray opines ***could*** create significant risk to other components of the Android stack, explicitly limits the application of that provision to works ***based on*** (*i.e.*, derivative works of) the Program in its opening sentence:

> You may modify your copy or copies of the Program or any portion of it, ***thus forming a work based on the Program***, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions: […].[21]

26. The sentence following the list of requirements cited by Ms. Murray confirms that "These requirements apply to the ***modified work*** as a whole." It goes on to clarify that simply distributing separate works together does not subject them to GPL obligations:

> If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then ***this License, and its terms, do not apply to those sections when you distribute them as separate works.*** But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.[22]

As a practical matter, this means that distributing the source code for parts of the Android framework together does not mean that one part is "tainted" by the open-source license requirements of another independent and separate but aggregated part. Rather, distributing the source code for various different components ("independent separate works in themselves")—for example, the libc and WebKit libraries and the Linux kernel—with the 37 OpenJDK-based

[19] *See* 17 U.S.C. § 101 (defining a "derivative work" as "a work based upon one or more preexisting works").
[20] GPL-2.0 § 0 (***emphasis*** added).
[21] GPL-2.0 § 2 (***emphasis*** added).
[22] GPL-2.0 § 2 (***emphasis*** added).

API packages does not mean that those other components become subject to the terms of the GPL-2.0-CE or GPL-2.0 license. Simply put, with respect to distributions of independent and separate but aggregated source code, the copyleft requirements of the GPL-2.0 license do not ***thereby*** extend to non-GPL-2.0-licensed software packages. I therefore disagree with Ms. Murray's opinion that "if works that are not derived specifically from the GPL-licensed code are distributed with the GPL-licensed code as a 'whole,' the GPL applies to those independent works."[23]

27. As if anticipating Ms. Murray's overly broad interpretation of the GPL-2.0 license's copyleft requirements, Section 2 concludes:

> Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective ***works based on the Program***. In addition, ***mere aggregation*** of another work ***not based on the Program*** with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License. [24]

In my experience and opinion, the open-source and closed-source software communities consistently interpret the GPL-2.0 license to extend its copyleft requirements only to distributed ***derivative works*** of the GPL-2.0 licensed software.[25]

---

[23] Murray Report ¶ 153. Related to the issue of what constitutes a "derivative work," Ms. Murray criticizes my statement that "On Dec. 24, 2015, Google publicly released source code for a modified version of the Android Runtime. That version is **derived from 37 Java API packages** from the OpenJDK project that are licensed under the GPL-2.0-CE license in place of the 37 API packages previously derived from the Apache Harmony project that were licensed under the Apache-2.0 license." Murray Report ¶ 165 (**emphasis** added by Murray). To be clear, my use of the phrase "derived from" in this context was not meant to equate to a "derivative work" in copyright terms, but rather that the updated Android Runtime includes classes from OpenJDK (which, as discussed in this reply, are each licensed separately). Tellingly, Ms. Murray implicitly acknowledges that treating the entire Android Runtime as a "derivative work" of OpenJDK is non-sensical, and that the opinions she expresses rest on a seemingly extreme interpretation of my statement as being "accurate" and "correct." Murray Report ¶ 165. Indeed, she otherwise concedes that the "that the OpenJDK-based code for the 37 API packages at issue reside in a folder in the master branch of libcore named 'ojluni.'" Murray Report ¶ 141. In short, although her factual understanding belies this tortured interpretation, Ms. Murray nonetheless offers strained opinions that rest on it.

[24] GPL-2.0 § 2 (***emphasis*** added).

[25] See, *e.g.*, Theresa Gue, *Triggering Infection: Distribution and Derivative Works under the GNU General Public License*, 2012 U. Ill. J.L. Tech & Pol'y 95 (2012) (available at http://illinoisjltp.com/journal/wp-

28. Relatedly, in paragraphs 169-171 Ms. Murray discusses several components of the Android software stack that she describes as being "used within the complete work," interacting with, or "using" the OpenJDK-based Java API packages, and "specifically designed to function as part of the whole Android stack." These include the Application Framework, Applications, and Bionic library.[26] Ms. Murray says that she understands that these components are "explicitly and materially dependent on the modified OpenJDK source code."[27] Yet Ms. Murray does not opine that the source code for such components form part of a ***derivative work*** of the source code for the OpenJDK-based Java API packages when distributed as part of the Android stack. Indeed, she provides no opinion that such relationships or dependencies result in a derivative work, and does not identify anything in the GPL-2.0 license terms that says that such relationships mean that these other elements of the Android stack somehow become tainted by the GLP-2.0 license. Instead, Ms. Murray merely hedges that such components "***may not be*** an 'independent and separate work'" and "***may need to be*** distributed under terms of the GPLv2 or GPLv2-CE license on order to ensure compliance with the license when Google distributes its source code for OEMs to use."[28]

29. The same problems appear in Ms. Murray's discussion of the Hardware Abstraction Layer ("HAL"). Relying on Dr. Schmidt, Ms. Murray states that the HAL has "no

---

content/uploads/2013/10/Gue.pdf) ("Copyleft is triggered by distribution and creating a derivative work.") Donald R. Robertson, III, *An Open Definition: Derivative Works of Software and the Free and Open Source Movement*, 42 New Eng. L. Rev. 339 (2008) ("The most important term of the GPL, the viral clause, can only be enforced against distributors of derivative works"); Derivative Works, Software Pluralism (available at http://www.law.washington.edu/lta/swp/law/derivative.html) ("The critical observation is that the requirement the GPL be applied to a new work is triggered when (1) a derivative work is created and (2) that derivative work is distributed."); Lawrence Rosen, *The Unreasonable Fear of Infection*, (available at http://www.rosenlaw.com/html/GPL.pdf) (concluding that the copyleft effect extends only to derivative works after acknowledging that the GPL states the effect extends to software "contains or is derived from" a GPL program); GNU.org, GPL FAQ.

[26] *See* Murray Report ¶¶ 169-171.

[27] Murray Report ¶ 170.

[28] Murray Report ¶ 170 (***emphasis*** added).

independent use outside of the Android platform, and that it is "explicitly and materially dependent on the Bionic library."[29] Even if true, this does not subject them to the terms of the GPL-2.0 license (with or without the Classpath Exception) when distributed together in source code form. They are independent and separate works of source code that are distributed together only when they are compiled into binary, executable code. As such, I disagree with Ms. Murray's opinion that "there is a significant risk that it too may need to be distributed under the terms of the GPLv2 or GPLv2-CE license in order to ensure compliance with the license when Google distributes its ***source code*** for OEMs to use."[30]

2. Distributing OpenJDK core libraries in source code form in aggregation with other parts of the Android stack does not impose copyleft obligations on the rest of the Android stack, *regardless* of whether the Classpath Exception applies.

30. Ms. Murray focuses on the word "executable" in the Classpath Exception. The word "executable" has been emphasized in the Classpath Exception language included below:

> Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License cover the whole combination.
>
> As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an ***executable***, regardless of the license terms of these independent modules, and to copy and distribute the resulting ***executable*** under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.[31]

The first sentence of the Classpath Exception explains that the exception is only necessary ***because*** "[l]inking this library statically or dynamically with other modules is making a

[29] Murray Report ¶ 171.
[30] Murray Report ¶ 171.
[31] Classpath Exception (available at http://www.gnu.org/software/classpath/license.html).

combined work based on this library." As discussed in my Opening Report, the Free Software Foundation ("FSF") supports the GNU project, including its underlying GNU General Public License (GPL) and Library/Lesser General Public License, and generally takes a broad reading of the GPL-2.0 license's copyleft effect.[32] While the FSF's interpretation of these licenses is not dispositive, that interpretation does provide some guidance as to the licenses' intended meaning and effect. Even under the FSF's broad view of the GPL-2.0 license's copyleft effect, it is the ***act of linking*** with GPL-2.0-licensed software that creates a "combined work" based on the library such that "the terms and conditions of the GNU General Public License cover the whole combination." ***But for*** the Classpath Exception (or some other exception), it is the FSF's view that ***if you link*** software licensed under the GPL-2.0 license with "other modules," the copyleft obligations from the GPL-2.0 license extend to those "other modules." Thus, the second paragraph of the Classpath Exception limits its permission to link independent modules with GPL-2.0-CE licensed software to produce an executable because, even under the FSF's broad view of the copyleft effect of the GPL-2.0 license, ***no exception is needed*** for source code—an exception is only needed ***if and when*** GPL-2.0-licensed source code is compiled and linked to create ***executables***.

31. This interpretation was confirmed in an article written by David Turner of the FSF and published on the FSF's GNU.org website:

> It has always been the FSF's position that dynamically linking applications to libraries creates a single work derived from both the library code and the application code. The GPL requires that all derivative works be licensed as a whole under the terms of the GPL, an effect which can be described as "hereditary." ***So, if an application links to a library licensed under the GPL***, the application too must be licensed under the GPL. By contrast, libraries

[32] Opening Report ¶¶ 50-57, and 68.

> licensed under the GNU Lesser General Public License (LGPL) may be linked to proprietary applications.[33]

Although this article is about the LGPL license, it confirms the FSF's position, stated in the opening to the Classpath Exception, that it is "linking" that creates a derivative work necessitating an exception to the GPL's copyleft requirements. The article continues by specifically applying the FSF's linking analysis to the Java programming language and concluding:

> ***When the application is compiled***, function signatures are checked against the library, ***creating a link***. The application ***is then generally a derivative work*** of the library. So, the copyright holder for the library must authorize distribution of the work. The LGPL permits this distribution.[34]

Thus, as Turner explains while specifically addressing the Java programming language, it is only after a Java application has been compiled and linked with a library that the application is then generally a derivative work of the library.

32. As explained in my Opening Report, the FSF position is also clarified in its published "Frequently Asked Questions about the GNU Licenses," which states: "[l]inking a GPL covered work statically or dynamically with other modules is making a combined work based on the GPL covered work. Thus, the terms and conditions of the GNU General Public License cover the whole combination."[35] The LGPL license echoes the FSF's position while explaining the purpose of introducing the weak-copyleft LGPL as an exception to the GPL-2.0 license's strong-copyleft requirements:

> ***When a program is linked with a library***, whether statically or using a shared library, ***the combination of the two*** is legally speaking a combined work, ***a derivative of the original library***. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of

[33] *The LGPL and Java*, David Turner (available at http://www.gnu.org/licenses/lgpl-java.en.html) (***emphasis*** added).
[34] *Id.* (***emphasis*** added).
[35] GNU.org GPL FAQ (available at http://www.gnu.org/licenses/gpl-faq.en.html).

> freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.[36]

The body of the LGPL license provides a more in-depth explanation of the FSF's position, describing how the source code for a program that links with an LGPL library is not a derivative work of the LGPL-licensed library, but that the ***act of linking*** that software with the LGPL library to produce an executable produces a derivative work:

> A ***program that contains no derivative of any portion of the Library, but is designed to work with the Library*** by being compiled or linked with it, is called a "work that uses the Library". Such a work, ***in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License***.
>
> ***However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library*** (because it contains portions of the Library), rather than a "work that uses the library". ***The executable is therefore covered by this License.*** Section 6 states terms for distribution of such executables.[37]

33. Thus, in the view of the FSF, the source code for the Android software stack—which is designed to work with the OpenJDK-based Java API packages, but has not yet been linked with the OpenJDK-based Java API packages—is not part of a derivative work of the OpenJDK-based Java API packages. Instead, as stated in the GPL-2.0 license, such OpenJDK-based Java API packages are merely distributed "in aggregation" with the Android source code:

> Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective ***works based on the Program***. In addition, ***mere aggregation*** of another work ***not based on the Program*** with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.[38]

[36] LGPL-2.1 preamble (***emphasis*** added).
[37] LGPL-2.1, §5 (***emphasis*** added).
[38] GPL-2.0 license, section 2 (***emphasis*** added.)

34. The FSF's position that unlinked source code distributed in aggregate with GPL-2.0 source code does not form part of a derivative work of the distributed GPL-2.0-licensed source code is consistent with opinions and practices in both open-source and closed-source software communities. Accordingly, in my opinion, Google's distribution of the OpenJDK-based Java API packages with the remainder of the Android stack in source code form does not require software ***other than*** the GPL-2.0-CE libraries to be distributed under the GPL-2.0-CE license.

3. Ms. Murray's overly broad interpretation of the copyleft impact of Google's distribution of the OpenJDK libraries has been expressly considered and rejected.

35. As I noted in paragraph 121 of my opening report, Bradley Kuhn, President of the Software Freedom Conservancy and FSF Board Member, opined that the Google's distribution of source code with code subject to a GPL-2.0-CE license would not virally taint the other parts of the Android source code in userspace:

> First, If you think the ecosystem shall collapse because "pure GPL has moved up the Android stack", and "it will soon virally infect everyone" with copyleft (as you anti-copyleft folks love to say) your fears are just unfounded. Those of us who worked in the early days of reimplementing Java in copyleft communities thought carefully about just this situation. At the time, remember, Sun's Java was completely proprietary, and our goal was to wean developers off Sun's implementation to use a Free Software one. We knew, just as the early GNU developers knew with libc, that a fully copylefted implementation would gain few adopters. So, the earliest copyleft versions of Java were under an extremely weak copyleft called the "GPL plus the Classpath exception". Personally, I was involved as a volunteer in the early days of the Classpath community; I helped name the project and design the Classpath exception. (At the time, I proposed we call it the "Least GPL" since the Classpath exception carves so many holes in strong copyleft that it's less of a copyleft than even the Lesser GPL and probably the Mozilla Public License, too!)[39]

[39] *Sun, Oracle, Android, Google and JDK Copyleft FUD* (http://ebb.org/bkuhn/blog/2016/01/05/jdk-in-android.html).

Notably, in making this assertion, Mr. Kuhn uses the term "weak copyleft" that Ms. Murray seems to reject, and also references the fact that others independently implemented the Java APIs at issue in this case.

36. Furthermore, Mr. Kuhn also addresses and rejects an argument raised by Ms. Murray in her rebuttal report that OEMs would not have taken the risk of using such a distribution, particularly given the fact that there was other source code subject to weak-copyleft requirements in the Android code base:

> So, how is incorporating Oracle's GPL-plus-Classpath-exception'd JDK different from having an Apache-licensed Java userspace? It's not that much different! Android redistributors already have strong copyleft obligations in kernel space, and, remember that Webkit is LGPL'd; there's also already weak copyleft compliance obligations floating around Android, too. So, if a redistributor is already meeting those, it's not much more work to meet the even weaker requirements now added to the incorporated JDK code. I urge you to ask anyone who says that this change will have any serious impact on licensing obligations and analysis for Android redistributors to please prove their claim with an actual example of a piece of code added in that commit under pure GPL that will combine in some way with Android userspace applications. I admit I haven't dug through the commit to prove the negative, but I'd be surprised if some Google engineers didn't do that work before the commit happened.[40]

37. Finally, for reasons similar to those discussed above and in my other reports, Mr. Kuhn concludes that redistributors of Android (e.g., OEMs) would not have been concerned about using source code licensed under a GLP-2.0-CE license because those same redistributors were already having to comply with the relatively more restrictive GPL-2.0 license for the Linux kernel:

> So, for Android redistributors, are there any actual licensing risks to this change? The answer there is undoubtedly yes, but the situation is quite nuanced, and again, the problem is not as bad as the anti-copyleft crowd says. The Classpath exception grants very wide permissions. Nevertheless, some basic copyleft obligations can remain, albeit in a very weak-copyleft manner. It *is* possible to violate that weak copyleft, particularly if you don't understand the

[40] *Sun, Oracle, Android, Google and JDK Copyleft FUD* (http://ebb.org/bkuhn/blog/2016/01/05/jdk-in-android.html).

> licensing of all third-party materials combined with the JDK. Still, since you must comply with Linux's license to redistribute Android, complying with the Classpath exception'd stuff will require only a simple afterthought.[41]

4. Ms. Murray's overly broad interpretation of the GPL-2.0-CE's copyleft requirements is inconsistent with Oracle's distribution of source code for the OpenJDK software.

38. Ms. Murray's conclusion that the distribution of the OpenJDK-based Java API packages in source code form "***could create a significant risk*** that additional Android source code, not currently covered by GPLv2-CE, would have to be distributed under that license"[42] is also inconsistent with Oracle's licensing, packaging, and distribution of the OpenJDK software. The OpenJDK FAQ, for example, states: "Each source code file is individually licensed" and instructs users to "look for the copyright header with the license information" in each file in order to identify the specific license that applies to that file.[43] A cursory review of source code files included in Oracle's OpenJDK source code distributions reveals that files licensed under the weak-copyleft GPL-2.0-CE license are regularly aggregated with files licensed under different open-source licenses (such as the strong-copyleft GPL-2.0 and permissive Apache-2.0 licenses) in the same OpenJDK package and, in some instances, even within the same package subdirectory.[44] Yet under Ms. Murray's interpretation, any further

---

[41] *Sun, Oracle, Android, Google and JDK Copyleft FUD* (http://ebb.org/bkuhn/blog/2016/01/05/jdk-in-android.html) (*emphasis* in original).
[42] Murray Report ¶ 21 (***emphasis*** added).
[43] As noted in my Opening Report, the Java FAQ site originally posted by Sun has been taken down and replaced since Sun's acquisition by Oracle. The original Java FAQ is still published in the OpenJDK section of the IcedTea project and is published available at http://icedtea.classpath.org/openjdk/java/faq.jsp.html. A snapshot of the original Java FAQ site taken June 4, 2011, by Archive.org is available at: https://web.archive.org/web/20110604004915/http://www.sun.com/software/opensource/java/faq.jsp. Contemporaneous versions were also produced in this case as GOOG-00000221 and GOOG-00000316.
[44] For example, downloading and extracting the openjdk-6-src-b27-26_oct_2012.tar source code archive published through the OpenJDK website reveals that the `jni_sparc` file (licensed under the weak-copyleft GPL-2.0-CE license) is included in the same subdirectory of the `Hotspot` source code package as many other files licensed under the strong-copyleft GPL-2.0 license such as `assembler_sparc`, and `icBuffer_sparc`. Moreover, Oracle incorporates numerous files licensed under the Apache-2.0 license, including those located in the \openjdk\ \jdk\src\share\classes\com\sun\org\apache\xml\internal\security\transforms subdirectory.

distribution of OpenJDK source code, even if identical to the OpenJDK source code distribution downloaded from the OpenJDK website, "***could create a significant risk***" of "***the whole***" subdirectory, package, or perhaps the whole OpenJDK distribution becoming subject to the copyleft requirements of the GPL-2.0 license. Accordingly, Oracle's own distribution of GPL-2.0-licensed source code files aggregated in the same subdirectory as source code files subject to the GPL-2.0-CE and Apache-2.0 licenses reveals that Ms. Murray's overly broad interpretation of the GPL-2.0 licenses' copyleft requirements is contrary to how Oracle itself distributes its OpenJDK software in source code form.

39. Relatedly, the fact that each separate, independent OpenJDK file qualifies as a "Program" under the terms of the GPL-2.0 license[45] undermines Ms. Murray's suggestion the OpenJDK libraries as a whole are licensed under GPL-2.0-CE.[46] In my opinion, because Ms. Murray's analysis and opinions rest on this erroneous understanding, they are flawed. Furthermore, Ms. Murray does not consistently apply the term "work" or "Program" as they are defined and used in the GPL-2.0 license.

E. Ms. Murray's summary of industry discussion and usage of GPL licenses is both misleading and erroneous.

1. Ms. Murray manufactures confusion regarding the GPL-2.0 and GPL-2.0-CE licenses.

40. In paragraphs 90-101 of her report, Ms. Murray cites several articles, blog posts, and responsive comments as support for her claim that "there is ample evidence that confusion and uncertainty about the meaning and impact of the GPL's copyleft terms is widespread in the

[45] GPL-2.0 § 0 ("This License applies to any program or ***other work*** which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. ***The "Program"***, below, refers to any such program or work, ..." (***Emphasis*** added)).
[46] *See* Murray Report ¶ 168 ("if Google modifies the Program (i.e. the OpenJDK libraries) and then distributes the modifications, Section 2 is triggered.")

industry."[47] However, as explained in the remainder of this section, none of Ms. Murray's citations support her claim of widespread confusion and uncertainty in the industry.

41. In paragraph 91 of her report, Ms. Murray quotes a blog post positing that the choice of a permissive license (such as the Apache-2.0 license) instead of a "viral" license (such as the GPL-2.0 license) does not meaningfully impact the adoption of that project in open-source and closed-source communities. The author then describes a series of project characteristics impacting adoption, which he refers to collectively as the project's "brand."[48] Notably, the post does not identify or discuss the Classpath Exception or GPL-2.0-CE license. Moreover, nothing in this post suggests that the author is confused by the Apache-2.0, GPL-2.0, or GPL-2.0-CE licenses. Indeed, the author explicitly acknowledges and addresses the "viral" nature of the GPL and LGPL licenses before concluding that such requirements do not meaningfully impact adoption of an open-source project.[49]

42. In paragraph 92 of her report, Ms. Murray quotes a reader's comment posted in response to the same blog post. The comment author suggests that, in comparison with the GPL and LGPL licenses, the Eclipse Public License (EPL) and Apache license are more (a) business friendly and (b) likely increase adoption and participation by "big corporations." The commenter then accuses Sun of strategically using the copyleft requirements of the GPL-2.0, GPL-3.0, and Classpath Exception "to ensure most depending companies come to them for a commercial license." [50] Ms. Murray cites the original blog post and the reader's comment as

[47] Murray Report ¶ 90.
[48] http://bill.burkecentral.com/2010/05/19/apache-damaging-to-open-source/.
[49] The author speculates that 90% of open-source consumers are unaffected by the copyleft requirements of the GPL and that "the number of people actually affected by the [LGPL's copyleft requirements] becomes a number that you can probably count on one hand." (available at http://bill.burkecentral.com/2010/05/19/apache-damaging-to-open-source/)
[50] Murray Report ¶ 92 (quoting Jilles Van Gurp, http://bill.burkecentral.com/2010/05/19/apache-damaging-to-open-source)

"just one example of many . . . highlighting the debate and confusion about the meaning of the various GPL license."[51] In my opinion, nothing in the original blog post or the commenter's response indicate that either author was confused by the Apache-2.0, GPL-2.0, or GPL-2.0-CE licenses. Again, both authors explicitly acknowledge and address the "viral" nature of the GPL and LGPL licenses. The authors simply reach different conclusions regarding the significance of the impact those "viral" requirements have on the adoption of an open-source project.

43. In paragraph 93 of her report, Ms. Murray cites to Stack Overflow, a website that describes itself as "a question and answer site ***for professional and enthusiast programmers***."[52] The cited question to which several programmers responded is: "What does 'GPL with Classpath exception' mean in practice?" A number of developers responded to the original inquiry attempting to shed light on the GPL-2.0-CE license. Notably, as quoted by Ms. Murray in her report, an early responder questions: "Shouldn't you be talking to your lawyer about this?" At best, this discussion might support a claim that at least some professional and enthusiast ***programmers*** find the GPL-2.0-CE license confusing. However, rather than relying on the legal opinion of professional and enthusiast programmers, it is my experience and opinion that OEMs would instead entrust analysis of such licenses and any related intellectual property concerns (such as the impact on the company's patent portfolio) to teams of internal and external attorneys and open-source licensing experts. As such, the Stack Overflow post cited by Ms. Murray does not, in my opinion, demonstrate confusion or debate among OEM decision makers.[53]

[51] Murray Report ¶ 93.
[52] http://stackoverflow.com/tour (***emphasis*** added).
[53] If anything, the post indicates that at least some professional and enthusiast programmers recognize the value (or at the least the necessity) of obtaining the advice of an attorney before relying on conclusions drawn about the GPL-2.0-CE license.

44. In paragraph 94 of her report, Ms. Murray cites to a 2008 publication by the Software Freedom Law Center (SFLC) entitled "A Practical Guide to GPL Compliance." In the cited section, the authors note that, despite common focus on the GPL's copyleft requirements, most of the SFLC's compliance challenges do not relate to companies failing to publish their proprietary software under the GPL license when required to do so. Instead, the guide indicates that companies generally run afoul of the GPL license by failing to provide or offer corresponding source code for distributed GPL-licensed binaries.[54] In my opinion, the experience described in the SFLC guide does not support Ms. Murray's claim that companies are confused by the GPL-2.0 or GPL-2.0-CE licenses. Indeed, the fact that companies regularly include GPL-2.0-licensed software in their commercial products but rarely combine their proprietary software with the GPL-2.0-licensed software in a manner that requires the publication of their proprietary code under the GPL-2.0 license leads me to believe that companies do indeed understand the copyleft requirements of the GPL-2.0 license and, specifically, understand how to combine their software with GPL-2.0-licensed software ***without*** subjecting their software to the GPL-2.0 license's copyleft requirements. Moreover, in my experience, companies failing to meet the GPL-2.0 license's requirement to publish corresponding source code often understand the requirements of the GPL-2.0 license but haven't yet implemented effective policies and practices that ensure that the requirements of open-source licenses are being met.

45. In paragraph 95, Ms. Murray quotes open-source attorney, author, and commentator Heather Meeker, who described her experience of corporate focus on open-source risk shifting away from infringement concerns (e.g., that the licensed open-source

[54] Murray Report ¶ 94 (citing https://www.softwarefreedom.org/resources/2008/compliance-guide.html).

software infringes third-party IP rights) and towards concerns that "their own particular use of the open source code does not comply with the requirements of the open source license applied to the code."[55] In my opinion, the experience described by Ms. Meeker does not provide evidence of "confusion and uncertainty about the meaning and impact of the GPL's copyleft terms."[56] To the contrary, as Ms. Meeker explains, companies are increasingly *focusing* their attention on whether their use of open-source software meets the requirements of applicable open-source licenses. Ms. Murray seems to suggest that such increased focus on "license compliance" is indicative of corporate confusion; however, in my experience and opinion, increased focus on the requirements of applicable open-source licenses by companies typically leads to better understanding of the open-source licenses rather than confusion or uncertainty.

46. In paragraph 96 of her report, Ms. Murray cites an alleged "uncertainty about the scope and interpretation of the various GPL licenses" as the reason some companies have been shifting their publication of proprietary software under an open-source license away from copyleft licenses (such as GPL and LGPL licenses) and towards permissive licenses (such as the Apache-2.0 license). In so doing, Ms. Murray seems to disregard the fact that Linux (which is subject to a GPL license) continues to be a very widely used open-source distribution. For example, a 2014 Linux Foundation report indicated that Linux was the primary enterprise cloud platform, with 75% of the market.[57]

47. Ms. Murray cites to several blog posts and published surveys in paragraphs 96-101 as support for her conclusion that companies are selecting permissive open-source licenses due to confusion and uncertainty regarding the copyleft requirements of the GPL-2.0 and GPL-

[55] Murray Report ¶ 95 (citing Heather Meeker, "Open Source and the Secrets of Commando Due Diligence," 44 Tex. J. of Bus. L. 561, 562.)
[56] Murray Report ¶ 90.
[57] http://www.linuxfoundation.org/publications/linux-foundation/linux-end-user-trends-report-2014.

2.0-CE licenses. In my opinion, Ms. Murray misinterprets or misunderstands the authors' explanations for the alleged corporate preference for permissively licensed open-source software.

48. Notably, not one quotation included by Ms. Murray in paragraphs 96-101 identifies confusion or uncertainty as a contributing factor to the alleged corporate preference for permissive open-source licenses. Instead, each of the references connect the selection of a permissive open-source license with ***increased adoption*** of the open-source project by corporate consumers of the licensed software who are often wary of potential copyleft requirements. Accordingly, contrary to Ms. Murray's claim that "confusion and uncertainty" lead to corporate preference for permissively licensed software, the cited references attribute the allegedly shifting preference for permissive licenses to companies' ***understanding*** of, and seeking to avoid the requirements of, the GPL and LGPL licenses.[58]

49. Notwithstanding the alleged shift away from the GPL and LGPL licenses and towards permissive open-source licenses, I also note that Oracle continues to publish many of its most popular open-source releases under the weak-copyleft GPL-2.0-CE or the strong-copyleft GPL-2.0 or AGPL-3.0 licenses:

[58] See, e.g., Murray Report ¶ 96 (quoting Alfresco CTO John Newton: "Although we were quite happy with the (RedHat JBoss) jBPM engine, ***it's LGPL license was preventing us from OEMing Alfresco to larger software companies that were concerned about any open source license with the letter G in it.***"); ¶ 98 (quoting OpenLogic Logic VP Kim Weiss: "enterprises tend to avoid the GPL in favor of the Apache license or other liberal licenses. […] These companies often try to avoid using open source licenses ***due to concerns about copyleft requirements and the impact on their own intellectual property***.") ¶ 100 (quoting 451 analyst: "***We typically see organizations focusing on*** some licenses that they approve for internal users or in products and licenses they do not support or condone. This is often the GPL, which is considered less permissive than other popular open source licenses, such as Apache Public License or Eclipse Public License. ***The requirements of the GPL,*** compared to a more permissive license, ***'are still perceived and truly are in some cases more onerous to the user, especially if it's a large enterprise that has high sensitivity around intellectual property***, both others' and its own."); (So that's the reason that these copyleft licenses – as they're called in a more general way – in the general pocket, are called the viral licenses and that's what everybody is scared of. So all of a sudden I'm writing this code, I have my intellectual property of doing some cool stuff and all of a sudden I find out that I am being using GPL license and now all of a sudden everything is out in the open and that's what scares a lot of enterprises.") (***Emphasis*** added)

Oracle Product	Oracle OSS License(s)
OpenJDK	GPL-2.0; GPL-2.0-CE;
MySQL	GPL-2.0
BerkeleyDB	AGPL-3.0
Oracle Linux	GPL-2.0

2. Ms. Murray conflates the GPL-2.0 and LGPL licenses with the GPL-2.0-CE license.

50. As Dr. Kemerer did in his opening report,[59] Ms. Murray repeatedly conflates the GPL-2.0 license with the GPL-2.0-CE license. The previous section identifies several references cited in paragraphs 91-101 that address GPL or LGPL licenses but do not identify or discuss the Classpath Exception or the GPL-2.0-CE license. The remainder of this section identifies additional examples of Ms. Murray conflating these licenses.

51. In paragraph 90 of her report, Ms. Murray quotes a law review article that states: "Historically, the GPL has been a difficult document to understand." That article does not identify or discuss the Classpath Exception or the GPL-2.0-CE license.

52. In paragraphs 125-129 and 131-134, Ms. Murray cites to comments made (a) by Google CEO Eric Schmidt, (b) by Android co-founders Andy Rubin and Rich Miner; (c) by a Google Developer Advocate, (d) on an Open Handset Alliance website, (e) in a Texas Instruments press release, and (f) in a report published by a market research company. These comments relate to the selection of the permissive Apache-2.0 license rather than a copyleft license (such as the GPL-2.0 license) for Google's contributions to the Android stack; they do not address concerns raised by Google's use of the OpenJDK-based Java API packages under the GPL-2.0-CE license. As explained in my Opening Report, my Rebuttal Report, and above, the copyleft requirements of the GPL-2.0-CE license are limited to ***modifications*** to the

[59] Kemerer Opening Report ¶¶ 160-182, addressed in my Rebuttal Report.

OpenJDK libraries and do not impose copyleft or other licensing restrictions on ***other software*** in the Android stack. Accordingly, the inclusion of the OpenJDK-based Java API packages in the Android stack does not restrict Google's license choices for the remainder of the Android software stack.

53. In paragraphs 129, 130, and 135 of her report, Ms. Murray cites to statements by Android co-Founder Andy Rubin expressing concerns over including software licensed under the strong-copyleft GPL-2.0 license in the user space of the Android software stack. In paragraphs 137-139, Ms. Murray cites to a presentation given at the Google I/O conference by Google technology program manager Patrick Brady. Paragraphs 137 and 138 discuss how the GPL-2.0 license covering the Linux kernel impacted Google's design of the Android architecture and how the Android architecture protects OEMs' proprietary software from the copyleft requirements of the GPL-2.0 license. As explained in my Opening report, my Rebuttal Report, and above, the inclusion of the OpenJDK-based Java API packages under the GPL-2.0-CE license does not raise the concerns identified with respect to the GPL-2.0 license. Furthermore, the fact that OEMs understood and accepted the strong-copyleft requirements of the GPL-2.0 license with respect to the Linux kernel suggests that they were sophisticated enough to understand and address the weak-copyleft requirements of the GPL-2.0-CE license within userspace.

54. In paragraph 131, Ms. Murray cites to a blog post on an Oracle website authored by an Oracle engineer and Java virtual machine architect describing the "top ten things [he] learned about Android and Dalvik VM" while attending a Google-sponsored conference for software developers. One entry doesn't cite specific sources but states:

> The first and main reason they give for using Harmony instead of OpenJDK is the GNU license (GPL). Cell phone makers want to link proprietary value-add

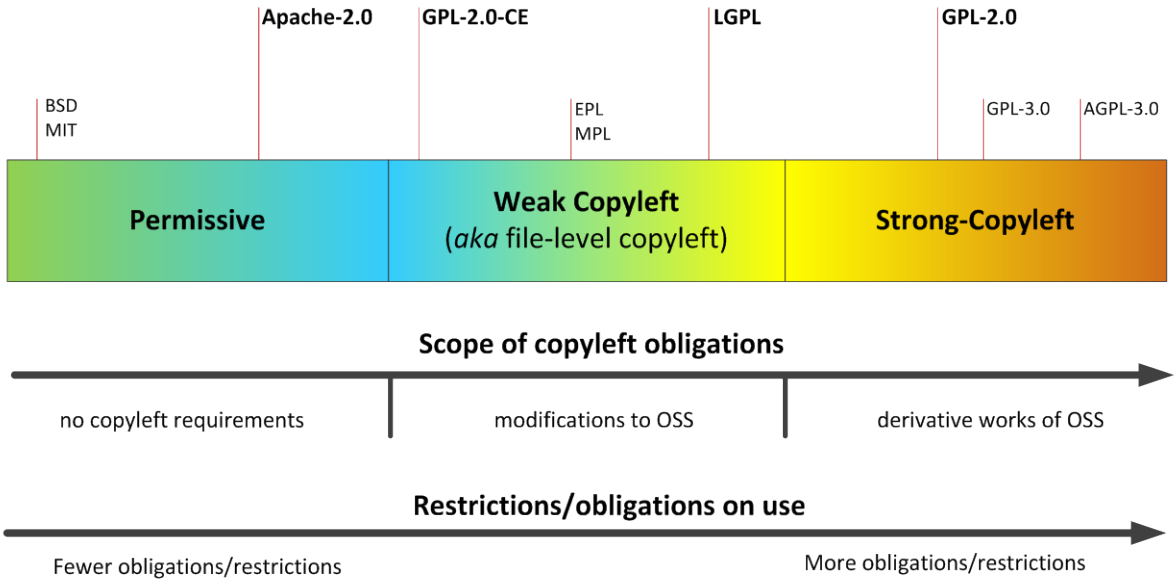
> code directly into the system (into JVM-based apps. and/or service processes), and they do not want to worry about copyleft.

Thus, the Oracle engineer's experience seemed to confirm Google's preference for the permissive Apache-2.0 license over the strong-copyleft GPL-2.0 license. The Oracle engineer confessed that he didn't understand the Classpath Exception; however, he was aware that the OpenJDK-based Java API packages were licensed by Oracle under the GPL-2.0-CE license and wonders whether he and the Google developers are in need of "some education" about the Classpath Exception.[60]

55. In my opinion the references cited by Ms. Murray do not support her claim that there is or was widespread confusion and uncertainty regarding the copyleft requirements of the GPL-2.0 or GPL-2.0-CE license. To the contrary, as detailed above, many of the references reflect corporate focus on and understanding of those requirements.

[60] "Perhaps there is some education needed here about the class path exception. (I know I don't understand it; maybe they don't either. And, their license wonks appear to have a well-considered preference for Apache 2 over GPL+CPE.)."

F. Ms. Murray's analysis disregards the fact that Android source code distributions have *always* incorporated open-source software included under *more restrictive* copyleft licenses than the GPL-2.0-CE license.



56. The diagram above plots popular open-source licenses (including those discussed in my Opening Report, my Rebuttal Report, and this Reply Report) from the most permissive (or least restrictive) open-source licenses to the most restrictive (or least permissive) strong-copyleft licenses. As explained in my Opening Report and Rebuttal Report, and as depicted in the diagram above, the GPL-2.0-CE is more permissive (or less restrictive) than the LGPL and GPL licenses that apply to other software packages already included in the Android software stack. In addition to the GPL-2.0-licensed Linux kernel, which provides the base of the Android software stack, numerous other Android stack components include one or more files provided under copyleft licenses that are *more restrictive* (or less permissive) than the GPL-2.0-CE license. The following table provides examples of such copyleft-licensed files:

Component/file names and links to source code	License
Portions of the elfutils component including, for example: platform_external_elfutils /libelf/elf64_getchdr.c platform_external_elfutils/lib/xmalloc.c platform_external_elfutils/libasm/asm_abort.c	Either: * LGPL-3.0; or *GPL-2.0.

Elfutils libelf components including, for example: platform_external_elfutils/libelf/dl-hash.h	LGPL-2.1
Portions of the webkit component including, for example:: platform_external_webkit/Source/WebCore/rendering /ScrollBehavior.h	Either: * LGPL-2.1; * MPL-1.1; or * GPL-2.0.
Portions of the webkit component including, for example: platform_external_webkit/Source/JavaScriptCore/runtime/BooleanPrototype.h; platform_external_webkit/Source/JavaScriptCore/ runtime/NumberPrototype.h; platform_external_webkit/Source/ WebCore/platform/efl/KURLEfl.cpp; and platform_external_webkit//Source/WebCore/platform/network/soup/GOwnPtrSoup.cpp	LGPL-2.0

57. In paragraph 140 of her report, Ms. Murray opines that the scattershot of statements she cites to (which almost exclusively address concerns raised by licenses *other than* the GPL-2.0-CE): "***cause me to doubt*** that Google would have incorporated into the Android Runtime any GPL license, including the GPLv2-CE, at the time … ."[61] In doing so, Ms. Murray not only improperly seeks to apply concepts relating to the GPL and LGPL licenses and their distinct requirements to the GPL-2.0-CE license, but she also offers no explanation why Google would include within its Android distribution software under the ***more restrictive*** GPL and LGPL licenses but refuse to include software licensed under the ***more permissive*** GPL-2.0-CE license in that same distribution.

58. Ms. Murray also confuses Google's selection of the Harmony-based Java API packages as an indication that Google found all other alternatives (including the OpenJDK-based API packages) unacceptable.[62] Logic and argumentation scholars refer to such erroneously drawn inferences as the as formal, logical fallacy of "improper disjunctive syllogism" or "affirming one disjunct."[63]

[61] Murray Report ¶ 140 (***emphasis*** *added*).
[62] *See*, e.g., Murray Report ¶ 21 ("And contrary to Mr. Hall's (paragraph 27) and Dr. Astrachan's (paragraph 264) assertions, those same risks would have been present if Google had decided to incorporate OpenJDK-based code in 2007—which of course, they did not do."
[63] *See*, e.g., Robert Audi (General Editor), *The Cambridge Dictionary of Philosophy (Second Edition)*, 1995, p. 316. ("formal fallacy[:] an invalid inference pattern that is described in terms of a formal logic."; "***Improper disjunctive***

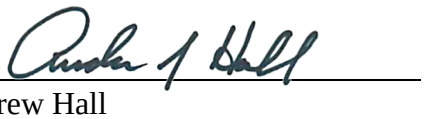
IV. DETAILED OPINIONS REGARDING DR. KEMERER'S REBUTTAL REPORT

59. In Section VIII of the Kemerer Rebuttal Report, Dr. Kemerer repeats (largely verbatim), the opinions expressed in Section VI.E of his opening report.[64] Rather than repeat the response I previously provided by way of my Rebuttal Report, I fully incorporate that analysis here.

V. CONCLUSION

60. For the reasons explained above, I disagree with Ms. Murray's conclusions regarding the scope of the GPL-2.0-CE license's copyleft requirements and that Google and/or its partners would not have accepted licensed alternatives to the 37 Java API packages in the 2007-2010 timeframe. To the contrary, it is my opinion that Google and its partners would not have had significant or lasting concerns in 2007 (or later years) if the 37 Java API packages in Android had been licensed under the GPL-2.0-CE license.

Executed on the 29th of February, 2016 in Novato, CA.



Andrew Hall

---

sylloligism (affirming one disjunct): *p* or *q*; *p* [therefore] not-*q*. 'Either John is an alderman or a ward committeeman; John is an alderman. Therefore, John is not a ward committeeman.'" (***Emphasis*** in original)).

[64] *Compare* Kemerer Opening Report ¶¶ 160-184 *with* Kemerer Rebuttal Report ¶¶ 230-255.

APPENDIX E – DOCUMENTS EXAMINED DURING REPORT PREPARATION

I examined the following documents, published materials, and websites in the course of preparing this report include:

- Documents cited in Appendix B to my opening report
- Documents cited in Appendix E to my rebuttal report
- Gwyn Murray's Rebuttal Expert Report and appendices, February 8, 2016
- Redacted copy of Douglas Schmidt's Rebuttal Expert Report and appendices, February 8, 2016
- Redacted copy of Chris Kemerer Rebuttal Expert Report and appendices, February 8, 2016
- Audi, Robert (General Editor), *The Cambridge Dictionary of Philosophy (Second Edition)*, 1995
- Deposition of Anwar Ghuloum, December 12, 2015
- Derivative Works Law Software Pluralism
- GNU.org GPL FAQ
- GOOG-00000221
- GOOG-00000316
- Gue,Theresa, *Triggering Infection Distribution and Derivative Works under the GNU General Public License*, 2012
- Kuhn, Bradley M., *Sun, Oracle, Android, Google and JDK Copyleft FUD*
- Robertson, III, Donald R., *An Open Definition Derivative Works of Software and the Free and Open Source Movement*
- Rosen, Lawrence, *The Unreasonable Fear of Infection*
- Turner, David, *The LGPL and Java - GNU Project - Free Software Foundation*